

NORTHERN ILLINOIS APPLE USERS GROUP

\$1.50

NOVEMBER 1982

THE HARVEST

VOL4-NO. 3

MEETING DATE November 13, 1982

10:00 am-1:00 pm Building A

IN THIS ISSUE

Club Officers, Members Aide, - - - - -	2
Editorial - - - - -	-3
Renewal Time is Here Again, Joseph Sobel - - - - -	4
Introducing Forth, Ray Mannarelli- - - - -	5
Perusing the Periodicals, Marlys Newcome - - - - -	6
Brain vs Computers, Skip Nieburger- - - - -	-8
Memory (and Other Things I Forget), Guy Lyle - - - - -	9
The Hotline Connection, Don Larson- - - - -	12
My Son and The Apple, Lloyd Stoner- - - - -	13
Digesting the Apple (Without Swallowing the Seeds), Robert D. Steinberg- - - - -	14
Talks of Interest - - - - -	16
News of Interest- - - - -	16
Club News - - - - -	23
Unclassified Ads/SIG News- - - - -	-18
Club News- - - - -	-19

CLUB OFFICERS

President-	Eric Stral	312-885-1941
Vice Pres-	Mike Robins	312-593-2709
Treasurer-	Joe Sobel	312-398-1826
Secretary-	Mary Rosemann	312-882-7938
Librarian-	Jim Pfeiffer	312-289-7311
SIG Co-ordinator-	John Hoffmann	312-998-0164
Software Chairman-	Rob Stewart	312-537-0544
Hardware Chairman-	Paul Stadfeld	312-359-2378
Program Chairman-	John Hoffmann	312-998-0164
Steering Committee-	Ted & Mary Rosemann	312-882-7938

Club Adresses:

MEMBERSHIP, etc.—BOX 787,
Palatine, IL, 60067

NEWSLETTER—Terry Tufts,
1015 S.Ridge Ave,
Arlington Heights, IL,
60005.
312-392-7735

BULLETIN BOARD—312-295-6926 24 hours
SYSOP-Dave Alpert

Membership is open to all. Dues are \$12.00 annually with a one time initiation fee of \$5.00 at the time of admission. Membership applications are available from the club Secretary at the meetings or by mail.

NIAUG NEWSLETTER is published by the Northern Illinois Apple Users Group and edited by Terry Tufts. Contents may be reproduced by other Apple Clubs for their newsletters or bulletins except where specifically reserved by the author. Permission to copy these articles may be obtained by writing the author through our editorial offices. Please indicate Northern Illinois Apple Users Group, HARVEST, is the source of any material reproduced. Newsletter exchange is welcomed. The HARVEST recognizes that APPLE and APPLE manufactured products are trademarks of the APPLE Computer Company.

MEMBERS AIDE

The members listed below have volunteered to answer questions from club members who need a 'HOT LINE' type answer that can be answered over the telephone. Please try to be brief when you call as a courtesy to them. Their names and phone numbers are listed below. The numbers after their names represent their special talents. PLEASE-NO CALLS at dinner time or after 10 pm.

Paul Stadfeld	312-359-2378	1,2,3,4*
Guy Lyle	312-438-7941	1,2,3,5,6
Earl Allen	312-837-9259	1-6,*
Ted Rosemann		
Mary Rosemann	312-882-7938	2,3
Joel Kurasch	312-677-8358	8,*
Leon Alexander	312-725-5309	11,*
Ken Nestle	312-620-7745	2,3,8
Herb Schulz	312-968-6927	8
Jim Bradshaw	312-881-7000	7
Rich Lundeen	312-420-8468	2,3
Eric Stral	312-885-1941	9,10
Joe Sobel	312-398-1826	9,10
John Hoffmann	312-998-0164	12
Joe Zeinz	312-526-0575	1-6

Mach. Lang=1	Int. BASIC=2
Applesoft=3	Hardware =4
Arrays=5	DOS =6
Education Software=7	Pascal =8
Business Software=9	Visicalc =10
Investment Software=12	
Z80 Card=11	*=1-6 inclusive

Numbers appearing with *, indicate the person is especially qualified in those items marked as well as items 1-6.

CORRECTIONS/CHANGES OF ADDRESS

Corrections/changes of address must be sent to the club secretary. Mailings are by bulk presorted third class mail. Any incorrect addresses will usually result in missed issues.

SUBMITTING ARTICLES

Handwritten articles are acceptable but articles on disk are preferred. Articles on a disk should use one of the following word processors: Easywriter, Applewriter, PIE or Magic Window. Use a 0 right margin and a line length of 47 characters for Easywriter and 41 for all others. Fill and right justify all text. Also supply a hard copy of the article for proof reading purposes. If your article includes a program listing please submit it on disk unless you have a printer that uses carbon film ribbon. All disks will be returned at the meeting unless requested to do otherwise.

EDITORIAL

by Terry Tufts

We are seeing some interesting forces at work in the marketplace. Most of the popular micro-computers on the market all use different operating systems. This has led to software that cannot be transferred between different computer brands. With 16 bit computers being added the picture is further complicated.

It seems pretty clear that a substantial number of manufacturers are going to play follow the leader with IBM by using the 8088 micro-processor and utilize at the very least the MS DOS and CPM 86 operating systems or some close variants. The easy and relatively safe way is to follow a company like IBM. Unfortunately this leads to stagnation of ideas and limits creativity. (If we had to depend on IBM to invent the Personal Computer we would still be waiting.)

The very act of creating a new and better product means that by its very nature it will not be "standard". That is not to mean that it may not accomodate an industry standard. By this I mean it still is possible to transport and run software on a foreign machine if enough forethought is used and there is an economic justification to do so. This can be done either by hardware add-ons such as the Z80 card in the Apple or by some form of emulation such as the Apple III uses to run some Apple II software or by specialized compilers.

Unfortunately, up until now, there has been no economic demand for compatibility and the manufacturers have taken a consumers be damned attitude.

Fortunately as the industry matures there are signs that this attitude is beginning to disappear.

One light at the end of the tunnel is SofTech Systems who have developed and are marketing the UCSD p-System which allows software to be transported between all the major micro-processors on the market.

Osborne Computer, Texas Instruments and Phillips Industries, NCR and Sage have adopted the UCSD p-System approach and a number of developers are announcing systems also using this system.

To further aid the process SofTech, has recently demonstrated its Universal Medium concept which allows floppy disks from one machine to be read by disk drives of several

different brands. While undoubtedly this will have some limitations, it is one more innovative step in making programs universal. As the number of users increases licensing costs decrease. This already is occurring. So it is now feasible for most software developers to use the UCSD p-System to have a portable program. Any serious software developers will be proceeding with peril if they ignore these developments. We see this as positive progress in computer evolution.

PASCAL ANYONE?? It appears that the Pascal language is becoming one of the dominant computer languages. Apparently Apple developing all its software. Pascal is on the verge of being a requirement for College level credit/admission. This alone will multiply its usage tremendously. If you are a beginning programmer you couldn't go very far wrong by starting with Pascal.

Speaking of Pascal. If you want to know about the latest developments in Pascal you will be interested in a relatively new publication that just passed across my desk. It is called PASCAL Market News. It is published 6 times per year. It is type set and tastefully laid out. Issue 3 consists of 20 pages with only couple of ads. The official subscription rates are \$20 per year but for a short time you can get an promotional subscription for \$11/year with some kind of credit for getting others to subscribe. It is well written, very informative and they will probably be getting my money. If you want more information you can reach them at PASCAL Market News, P.O. Box 5314, Mt. Carmel, Ct 06518. (203)288-0283.

Between These Pages

This month we welcome articles from several new contributors. It is nice to see people stretching themselves and taking a small risk. We try to make this as easy and painless as possible and we hope you enjoy their contributions.

Our first article on Forth has been written by Ray Mammarelli. I found it very interesting and it nicely complements the presentation given by Paul Stadfeld at the last meeting. There are a number of people in the group who know Forth and perhaps we can get a SIG started. If you are interested, talk to Ray. He doesn't want to run a SIG but would be a focal point to get one started if there is any interest.

RENEWAL TIME IS HERE AGAIN

By Joseph Sobel, Treasurer

Well, its renewal time again. As you all must be aware by now, the cost of everything is rising today. NIAUG has not been immune to the ravages of inflation any more than the rest of the economy. Ever since NIAUG was founded over three years ago, there has not been an increase in the annual dues. However, we now find ourselves faced with the need to raise the annual membership fee in order to maintain our ability to serve the ever increasing ranks of our membership. The increase, to \$24.00, has been necessitated by the general increase in prices experienced by all of us. In addition, there has been little if any change in our financial position since the beginning of the year. But, more importantly, there are many future contingencies which may require substantial amounts of cash in order for us to maintain our present activities. These needs include the possible increase in rental rates for meeting facilities necessitated by a move to a new location and the related costs for audio/visual equipment which may be required. The group's records are now being maintained on a general ledger system on the Apple. For those of you who are interested in the group's finances, they will be reviewed at each planning meeting on the Tuesday evening following the general meeting.

The good news is that your increased dues will continue to provide you with all of the activities and membership benefits which have been such an important factor in sharing our knowledge, or lack thereof, of the Apple, including:

1. Monthly newsletter
2. Monthly meetings
3. Help line
4. Disk library
5. Paper library
6. Special interest groups (SIG's)
7. IAC representation

Thanks to the conscientious work of our secretary, Mary Rosemann, we have an up to date membership list available for use in this year's renewal process. This year, all members will have to complete an application form. This will provide us with documentation so we can avoid any misplaced renewals. In addition, the application has been shortened to facilitate completion. A renewal application and a postage paid return envelope will be mailed to all present members during late October or early November. Please return your application and check in the envelope provided.

All members who have not renewed by the January meeting will be dropped from the membership list during January. If these members wish to join subsequent to the January meeting, they may do so by submitting an application together with the required dues, as shown on the application, for the balance of the year plus the first year initiation fee. We are sorry, but we will be unable to provide back issues of the Harvest which may have been missed during the time their membership had lapsed. If available, these issues may be purchased at the meetings.

We would like to have as many members as possible renew early in order to avoid any unnecessary confusion toward the end of the year when we are all involved in the holiday rush. With your help, this year's renewal process should run quite a bit smoother than last year's. If you send in your renewal, but you do not continue to receive your Newsletter, please contact Mary Rosemann as she maintains the mailing list. If your check has not cleared the bank and you believe it may be lost, please contact me so we can track it down. We are looking forward to another year of informative demonstrations and to the same type of knowledge sharing that has made NIAUG great in the past.

INTRODUCING FORTH

by Ray Mannarelli

Many Apple users have heard of the Forth language and of the benefits of its fast execution speed and small memory requirements. Unfortunately, discussions of Forth are usually couched in language unique to Forth and tend to raise more questions than answers. In this brief introduction I hope to explain a few of Forth's basic characteristics and their implications for Forth's practicality.

Perhaps the most central concept inherent in Forth is its use of--what's termed--a dictionary. When Forth resides in your computer, its memory is loaded with the Forth dictionary. This dictionary consists of a string of many short individual programs--or "words"--that make up the sum total of the Forth language. The words are arranged so that the simplest functions--fetch a byte from memory or perform an "AND" function--appear in low memory; and progressively more complex functions--IF statements and I/O commands--appear in higher memory. The very simplest words are coded in machine language but the rest consist of subroutine calls to simpler words in the dictionary. For example, the program to multiply a given number by 2 first jumps to the routine (in lower memory) to place the number two in a register and then jumps to the function (also in lower memory) to multiply two numbers together. In this way, more complex routines consist of subroutine calls to simpler routines.

The biggest implication of this building-block approach is that Forth programs typically take up less memory space than the equivalent Fortran or Basic program. If we write a program in Forth in which uses the multiply by two function on five separate occasions, our Forth program will have the address of the multiply by two function in it at five places in the program. If we write and compile the same program in Basic when look at the compiled program we will have the entire code to multiply a number by two repeated five times in the length of the program. In short once Forth defines a function, it need exist only in one place in memory; many other languages

must repeat the entire code for the function everywhere in memory it is used.

A second property of Forth that sets it apart from most other high level languages is its use of a data stack. The usual way to think of the stack is like a spring-loaded stack of dishes in a cafeteria line--when you take a plate off the top the others move up one position, and when you put one on top the entire stack moves down one. With Forth's data stack when you put a number on top of the stack, all the other numbers on it move down one position and when the top number comes off the others move up one spot.

Almost all number manipulation in Forth involves the top two numbers on the data stack. When Forth is told to add two numbers it removes the top two numbers from the data stack, adds them and puts the sum on top of the stack. If we want to know if two numbers are equal, we put one on the stack, then the other, then call a routine labeled with an equals sign (=). This function removes both numbers from the stack, compares them and puts a zero on the stack if they're not equal or a 1 if they are equal.

There are two practical implications to this heavy use of the data stack. The first is a strong deemphasis in Forth on the use of labelled variables, i.e. SUM for a running total or AVRG for the mean of a set of numbers. By leaving heavily-used variables on the data stack we eliminate the need to store them in a special area of memory along with an associated symbolic label for each--more memory saving.

Another benefit of the data stack is that it provides a simple way of transferring variables from one routine to another within a Forth program. By contrast, in Fortran to transfer a set of variables from the main routine to a subroutine, the calling statement must have a list of variables in it that match a list of analogous variables in the first statement of the subroutine. With Forth, both the calling and the called routine assume that the numbers to be operated on are on the stack and so all the matching of variables in the previous example is eliminated.

This is a very brief introduction to FORTH. I hope it will challenge you to learn more about it.

* * * * *

PERUSING THE PERIODICALS

by MARLYS NEWCOME

CREATIVE COMPUTING VOL 8 #10 SEPT/82

Author David H. Ahl
 Title APPLE ARCADE GAMES
 Page 66
 Subject A review of 11 new games

Author Molly Watt
 Title WHAT IS LOGO?
 Page 112

Author Donald T. Piele
 Title HOW TO SOLVE IT- WITH THE COMPUTER
 Page 140
 Subject International Computer Problem Solving Contest II results

Author Barry Mc Connell
 Title THE HANDICAPPLE
 Page 186
 Subject A low cost braille printer

Author Edward Mitchell
 Title INTRODUCTION TO DATA STRUCTURES
 PART 2
 Page 210
 Subject Searching techniques- This article explores several data structures- lists, stacks, queues, and trees

Author David Lubar
 Title THE GRAPH PAPER PART 4
 Page 264
 Subject A discussion of collision detection and paddle control

This issue has many reviews of educational software packages

INTERFACE AGE VOL 7 #10 OCT/82

Author Carl Heintz
 Title BUYER'S GUIDE: INVENTORY CONTROL SOFTWARE
 Page 45

Author Dan W. Post

Title TELECOMMUTING: TOWARD AN EXTENDED OFFICE

Page 61

This issue contains reviews of VisiDex, PFS, Sensible Speller, and S.B.C.S. General Ledger version 4.0.

SOFTSIDE # 33

Author Brent Iverson
 Title SHAPE WIZARD
 Page 30
 Subject Create and manipulate shapes on the APPLE with this utility

Author Gary Schroeder
 Title DEFENSE
 Page 36
 Subject An arcade type game to type in and play.

This issue contains reviews of Graphic Processing System, The Complete Graphics System and Special Effects. There are also several articles about computer graphics.

COMPUTE! VOL4 #9 SEPT/82

Author Richard Cornelius
 Title BANISH INPUT STATEMENTS
 Page 56

Author Alan W. Poole
 Title TURTLE PILOT
 Page 84
 Subject Type in this program and have a Turtle Pilot Compiler in BASIC for the APPLE.

Author Keith Falkner
 Title PROCESS ANY APPLE DISK FILE
 Page 128

MICRO THE 6502/6809 JOURNAL #52 SEPT/82

Author Tim Osborn
 Title APPLE SLICES
 Page 47
 Subject "Hashing", a quick random-access

technique.

Author Phillippe Francois
 Title AMPER-POS FOR THE APPLE
 Page 51
 Subject A machine-language utility to
 extend Applesoft to include the
 POSiton function.

Author Barton M. Bauers
 Title COMPRESS- AN APPLESOFT OPTIMIZER
 Page 89
 Subject A machine-language utility that will
 reduce Applesoft program memory
 requirements by up to 30 per cent.

BYTE VOL7 #9 SEPT/82
 * * * * *

Author Robin Moore
 Title THE APPLE III AND ITS NEW PROFILE
 Page 92

Author Joseph H. and Jeffrey S. Taylor
 Title A COMPARISON OF FIVE COMPILERS FOR
 APPLE BASIC
 Page 440

Author Larry Korba
 Title TURN YOUR APPLE II INTO A STORAGE
 OSCILLOSCOPE
 Page 520

This issue has many articles about the use of
 computers by the disabled.

PERSONAL COMPUTING VOL6 #10 OCT/82
 * * * * *

Author Theresa Engstrom
 Title THE SMALL BUSINESS WAS UNDER
 FINANCIAL SIEGE
 Page 82
 Subject An accounts-receivable package
 put cash flow back on the track

Author Lee The
 Title MORE COMPUTER MUSCLE TO POWER THE
 MACHINE
 Page 88

Author Mike Barlow
 Title THE SMART HOUSE COMES OF AGE
 Page 112

Author Leon Starr
 Title PROGRAMMING FOR THE WRITTEN WORD
 Page 132
 Subject Second in a 4 part series about
 creating a simple word processor

POPULAR COMPUTING VOL1 #12 OCT/82
 * * * * *

Author Steve Ditlea
 Title VISICALC AND "VISICLONES"
 Page 48

Author George Stewart
 Title TK SOLVER
 Page 53
 Subject A review of this new number
 crunching program

Author John Edwards
 Title GAMES FOR THE APPLE
 Page 73
 Subject A review of 8 games

Author Ken Skier
 Title THE NEW STEP BY STEP
 Page 98
 Subject A review of this multimedia BASIC
 programming course for the Apple.

Author Henry Ledgard and Andrew Singer
 Title ELEMENTARY BASIC
 Page 120
 Subject The fourth in a series of articles
 excerpted from this book

CALL A.P.P.L.E. VOL5 #9 SEPT/82
 * * * * *

Author Cornelis Bongers
 Title EXECUTE STRINGS
 Page 9
 Author Ralph Swerdlow
 Title Printer protocol
 Page 17

Author Nelson R. Capes
 Title MONEY MANAGER
 Page 21
 Subject A complete program for you to type
 into the computer

Author R. D. Purves

Title TEXT FILE FAST LOADER

Page 39

Author Charles Kluepfel

Title ADVENTURES WITH A RAM CARD

Page 53

Author Eric Lambrecht

Title REMSAVER

Page 59

APPLE ORCHARD VOL3 #4 SEPT-OCT/82

Author Woody Liswood

Title LOCKSMITH AND THE INSPECTOR

Page 58

Author William Harvey

Title SECRETS OF PROFESSIONAL GRAPHICS

PART I -BEHIND THE SCENE

Page 64

Subject The first of three articles about graphics

Author Joseph J. Sobel and Barry D. Bayer

Title VISICALC TIPS AND TECHNIQUES

PART 2: PRINTING,SAVING AND ENHANCEMENTS

Page 74

BRAIN VS COMPUTER ©

by Skip Nieburger

Reprinted with permission from the Dental Newsletter

Have you ever wondered how complex the human brain is when compared with the modern computer? Sometimes when you consider calculation or sorting speed, the computer is far more advanced than the human mind...or so it seems. Actually the human brain and digital computer are different. The brain behaves like an analog computer responding to amounts of stimulation rather than the on-or-off nature of the widely used digital computer.

The human brain occupies anywhere from 300cc (child) to 2000cc (largest adult) of cranial capacity. No matter what the size of the head, there are

about 10 trillion (10,000,000,000,000) neurons (nerve cells) which weigh 3 pounds but consume 25% of the body's energy and oxygen. If you consider each neuron equivalent to one byte of computer memory, you possess in your own head, a 10 trillion byte computer. What is the power of this machine?

In order to comprehend you must put things in everyday perspective. let's use graphics. The best graphics computer system available is the earth bound computer that interprets the pictures sent back by the Voyager space craft (e.g. Saturn fly by). Each picture is made of a grid of 64,000 dots (even the little Apple II computer gives 53,000 dots per screen) which can be lightened or darkened by 256 levels of intensity. That's just one picture which takes approximately 3 minutes of computer time to form.

The brain is a bit quicker. Pictures (what you see) are registered every few nanoseconds...thousands per minute. These are placed into permanent memory and can usually be recalled by "remembering". Think of a particular moment of your life (a childhood experience, getting married, etc.) How many single frames would it take to recreate that episode if it were on film or videotape? Thousands, millions per second? Remember each frame of your mind's computer is infinitely more detailed(more dots) than the 64,000 dot matrix of the Voyager computer.

You can now get an inkling to the complexity of the human mind. Instant recall of billions of bytes of graphics alone....then consider sound, smell, pressure, humidity,etc. that is remembered at each moment coupled with the interpretations and emotions. Truly fantastic.

Of all the computers available, your 10 trillion byte random access memory mind is the cheapest and best...though it still has a few glitches and bugs.

MEMORY(and Other Things Which I Forget)

by Guy A. Lyle

(Copyright (c) 1982, Guy A. Lyle: May not be copied in whole or part without written permission of author)

Hello

Hello, fellow Apple nuts! Or is it Apple seeds? I started this series of articles a couple of years ago. In re-reading them, they are severely out-of-date. Believe it or not, Appledom has changed a lot in just a few years. So I am going to start again, re-writing as I go. The series of articles will cover as many aspects of computers, the Apple II in particular, as I have time to cover. The objective will be: To give the Apple II user an insight into the internal workings of the Apple II and microcomputers in general.

First a note about yours truly. I am, by profession and hobby, a "software engineer". That's a buzz word which means that I write computer programs which interact highly with the "hardware" of the computer. This more or less defines my point of view when I am discussing the workings of a computer. I tend to start from the inside of the machine and work my way out rather than, say, starting with Applesoft Basic and then working my way down to the finer points of the inner guts. If your sole interest is getting your machine to do your payroll or inventory, then you're probably not going to be too interested in what I have to say. But if you're interested in just how these machines do what they do, then follow along.

Pieces of the Puzzle

The Apple II is typical of all computers. I see computers as being composed of four

major parts: Input, output, processing unit, and memory. Many other authors

view them as three major parts, lumping the processing unit and memory together. I prefer to distinguish the two, although they generally go together hand-in-hand.

I suppose one could have a computer without any input or output parts. There would be no point, though. To be useful, the computer needs to take in information in order to process it. The Apple's keyboard is an example of such an input part. Output is also essential. Without it we couldn't get the processed information back from the computer. Your monitor screen is a sample of such an output device. Some devices handle both input and output, such as the disk drives. Data can be loaded into the computer from the disks (input). Data can also be stored on the disks for later use (output).

The Central Processing Unit, or CPU, of the computer handles all of the arithmetic and decision making. It is really the heart of the computer. The Apple II uses a "6502 microprocessor" for its CPU. "6502" is the model number. Some model numbers used by other computers are: 8080, Z80, 6800, 6809, 8086, 68000, and so forth. These different models all do basically the same thing. It is the way in which they do them that is different. Each has its advantages and disadvantages, which we may discuss at some other time. The term "microprocessor" indicates that the CPU comes as a single package rather than a whole bunch of pieces. The 6502 package can be found in the Apple just in front of the I/O slots. It is the large black package placed left-to-right, rather than front-to-back like all the other packages. The metal pins along each side of the package connect it to the rest of the computer. This type of package, by the way, is often called an "integrated circuit", IC, or sometimes a "chip" (the silicon chip is actually inside the black package).

The Memory Piece

The memory part of a computer does exactly that -- it remembers things. There are two general types of things which it must remember. These are (a) instructions or programs and (b) information or data.

While the CPU of the computer is very powerful, it is totally incapable of doing anything for itself. It must have a set of instructions to follow. We generally call these instructions a "computer program". The CPU finds its instructions in the memory part of the computer. When you load a new program into your computer, you are just putting a new set of instructions in the computer's memory. The instructions a computer follows make the computer take in information, process the information, and return the processed information to the computer user. The computer is simply an information processing machine.

The computer must have someplace to store the information while it is being worked on. Hence the computer's memory is called on again for this purpose. An example would be an address list data base program. Such a program would load all or part of the addresses into the computer's memory in order to modify or display them.

The computer's memory has often been described as being like a stack of pigeon holes or post office boxes. Each of these pigeon holes is called a memory location. There are a great number of memory locations within the computer. The amount of memory a computer has is measured in groups of 1024, or 1K. A computer with 48K of memory has 48 x 1024 or 49,152 memory locations. The amount of memory that a computer can use is limited by its CPU. The 6502 microprocessor in the Apple has a limit of 65,536 memory locations, or 64K.

Now someone is bound to ask, "If this is the case, what good are all those memory expansion boards for the Apple?", or "What good is it to buy a 128K memory expansion when the Apple can only access 64K?". The trick is that the 6502 can

only access 64K of memory at ANY ONE TIME. You can have access to more than 64K of total memory by switching your access to different areas of the expansion board's memory at DIFFERENT times. This is called "bank switching". It is much like a revolving door which opens into many different rooms. While it opens into many different rooms and provides you with a large amount of

space, it can only open into one room at a time. In order to access a different room you must revolve the door to that room, cutting off immediate access to the previous room. This technique has the advantage of opening up large amounts of memory for data storage. The disadvantage is that it takes computer programs which know how to perform the bank switching.

In order for the computer to keep track of which memory location is which, each memory location is assigned a number. These numbers are called "addresses", just like house numbers along a street (memory lane?). The lowest address is 0. The highest (for the 6502) is 65535, which is one less than 65K. This gives the total of 65,536 memory locations.

What can ya' tell us?

Basically, each memory location is a storage place for a number. The smallest number which can be remembered is 0. The highest is 255. In Basic there is a function which allows you to see what number is being stored in any memory location within the computer. This is the PEEK function. A simple way in which to use it would be to enter the command:

```
PRINT PEEK(3)
```

The computer will print the number currently being stored in the memory location whose address is "3". To see what is in memory location 32000, type:

```
PRINT PEEK(32000)
```

Applesoft and Integer Basic can also use negative numbers to refer to memory locations. The Apple doesn't have "negative memory". What happens is that negative addresses actually count down from the top of the Apple's memory. Memory location -1 is the same as 65535, -2 the same as 65534, -3 the same as 65533, and so forth. To find the

positive value of a memory location, add (really subtract) the negative value to 65536:

```

65536
-16384 Example negative address
-----
49152 Actual positive address
      What's My Type?
  
```

Back in the "old days" when Apples were still a relatively new thing, we were told that their memory limit was 48K. Were the dealers or Apple trying to cheat us out of our legal limit of 64K? Actually not. This 48K was the maximum of a certain type of memory in the computer called RAM memory. There are three types of memory locations in the Apple: RAM, ROM and I/O. The "missing" 16K is required by the Apple's ROM and I/O memory.

RAM stands for Random Access Memory. It is actually misnamed. It should really be called RWM for Read/Write Memory, but history has left us with the RAM misnomer. Its distinction is that its memory locations can be "read" and "written". Reading memory involves looking at a memory location to see what number is being stored there. Using the PEEK function causes a memory location to be read. The write function, on the other hand, causes a new number to be stored in a memory location. In Basic the POKE statement causes a memory location to be written. For example, the command:

```
POKE 768,12
```

causes the number 12 to be stored in memory location #768. Using the PEEK function you can verify that indeed 12 is now stored in this memory location. If you then give the command POKE 768,55, you will change the number stored in memory location #768 to 55. The old number, 12, is lost forever. This is the basic operation of RAM type memory: You can read the number stored in any memory location, and you can change the number being stored there.

One of the problems with RAM type memory is that it forgets everything when the power is turned off. This is the reason that you have to reload a program when you turn your Apple II back on.

The lowest 48K memory locations within the Apple II are reserved for RAM type memory.

ROM, or Read Only Memory, only allows the numbers stored within it only to be read. ROM memory locations cannot be changed. For example, try to PRINT PEEK(63488) [or PRINT PEEK(-2048) for you Integer Basic fans)]. You should get a "74". This is a piece of you ROM memory. If you try to POKE 63488,55, the number 74 will still be stored in this memory location. It simply refuses to be changed. In fact, the only way to change it is to replace the package in which this memory location resides.

The big advantage of ROM type memory is that it does not forget when power is turned off. The same information is still stored there when the power is turned back on. The Applesoft Basic (or Integer Basic) that came with your Apple II is stored in part of this ROM type memory. There are also some general purpose type programs in the Apple's ROM that form a set of programs called the Monitor. The ROM memory occupies the highest 12K memory locations in the Apple II.

Another feature of the Basic language is the CALL statement. This statement tells the computer to execute a program in the computer's memory at a given location. For example:

```
CALL 65338
```

will perform a program which makes a tone and the Apple's speaker.

This leaves us with the 4K of "I/O" type memory in the Apple II which must lie between the RAM memory and the ROM memory. The term I/O means "Input/Output". I/O memory locations, however, are not really memory locations. Instead they are the means by which the 6502 microprocessor communicates with the outside world. Consider the following program:

```

10 V = PEEK(-16287)
20 IF V > 127 THEN PRINT "PUSHED"
30 IF V < 128 THEN PRINT "RELEASED"
40 GOTO 10
  
```

While this program is running, push and release the game button on game paddle #0. Notice that your program can detect the state of the game paddle button. When the value the computer finds in this memory location is greater than 127, you are holding the button down. If not, then you have released the button.

Another special I/O memory location is -16384. Consider the following program:

```
10 C = PEEK(-16384)
20 IF C < 128 THEN 10
30 PRINT CHR$(C);
40 POKE -16368,0
50 GOTO 10
```

While this program is running, type characters at the keyboard. Voila! Memory location -16384 is where the Apple senses the keyboard! When the value in this location is 128 or greater, then it is holding a number which represents the key just hit. Applesoft's CHR\$ function turns this number back into its equivalent character. What's the extra POKE command in line 40? Well that's the way the computer tells the keyboard that it has read the character from the keyboard. Just accessing this location either by reading it or by writing it forces the value in location -16384 to fall less than 128. The value will not become 128 or greater again until another key is struck.

As you see, I/O memory is not really memory. It is simply connections to the outside world. The peripheral slots at the rear of the Apple II, where you plug disk drives, printer cards, and so forth in, are all accessed via this I/O memory area. The basic connections (memory locations) used by the Apple II are described in Chapter #1 of your Apple II Reference Manual.

At Last,
I Though He'd Never End

If you haven't developed a headache yet, then good for you. If you have, welcome to the crowd. If you are new to computing then many of the things I have talked about will be confusing. This is

100% normal. All of us started this way, and no one of us learned it all at once. Learning computer "stuff" is a matter of constant exposure -- It slowly rubs off on you after repeated experiences. One thing does help, though. Experiment. Try things out. You can't really hurt your Apple in any way. Contrary to rumor, it does not have a self-destruct command. Take the floppy out of the disk drive and have fun. Only your computer will know your mistakes: It won't tell and you can cuss it out at any time!

I hope to continue these learning experiences. Your feedback will, of course, help. In general I will try to build up, using this article as a base, a first stepping stone.

May the fates of computerdom be with you....

* * * * *

THE HOTLINE CONNECTION

by Don Larson

I had been a member of NIAUG for some time, before I ran into a problem in programming with Applesoft. I was trying to find the method employed to save a 'Hi-Res' picture. Perhaps this seems very basic (pun intended), but at the time, in my haste I just couldn't find the information in the manuals. I knew that the club Hotline was available to me for just such a situation, but I felt foolish calling someone for such a little problem.

Well, after much thought I finally got enough nerve to place a call. As the phone rang at its destination I wondered how to start the conversation. My thoughts were interrupted by the response of the person picking up the phone with a courteous "hello". I then replied in kind and proceeded to explain who I was and why I was calling. The person expertly handled my question (not once did he even hint at being annoyed with 'not another one of those calls', which relieved me of the idea that he would be rolling on the floor with laughter). We talked for a few minutes longer before ending the conversation. Afterwards, I realized how easy it was to request assistance and then, how glad I was that help was there when I needed it. At this time I wish to thank the people that are a part of the Hotline for a job well done. They deserve it.

My Son and the Apple

by Lloyd Stoner

Dear Terry...

I liked your guest editorial, "14 Reasons...". It spurred my resolve to write an article. One problem that it did not answer was: 15. I don't know where to start. I guess I will have to start by saying this is a letter about my son and his Apple.

We always had an electric typewriter. John discovered it at an early age, and spent a lot of time using it. But the typewriter wasn't enough. You see, making mistakes on a typewriter means you have to pick up an eraser to remove the offending spots. This became a problem for John.

John has an affliction called Dystonia Musculorum Deformans. Writing legibly is a problem. His muscles don't respond properly. Even using an eraser with a typewriter is too much of an interruption to the flow of ideas. We moved on to the computer. It wasn't an easy decision to make. It was just a course of action that was necessary.

Eighth grade calls for writing themes and book reports and special projects. He had to be able to perform with the class, not only in competition for grades, but to meet the standard of his own intellect. High School would be more demanding. Apple solved this immediate problem and if it were to give John an insight to computers as a lead on a future livelihood, so much the better. We went ahead to set up the family computer center.

I was little help from the start. After all didn't I have to go to work every day, check the newspapers, fix broken faucets and generally be the head of the house? While I was going through all this B--- S---, John was following directions, learning how to "boot" things and discovering "menus" that included things that weren't meant to eat. My ego was doomed to deflate when I asked the 13 year old for help and was told..."You'll just have to read the manual Dad."

Earlier John had been interested in codes and cryptology. He had burned out a typewriter ribbon making up codes for "club" of buddies in grade school. Now, he took off on programming. He took a print out of all the prime numbers up to 10,000 to his eighth grade math teacher. Book reports written with Apple Writer made a hit with the English teacher. He was getting encouragement everywhere. That and having the gadget upstairs was enough of a push. He didn't need any encouragement from me, except or getting him to write a letter to his Grandmother.

We heard about NIAUG and went to a meeting. In spite of the fact that I didn't know what was going on, John did, and wanted to go back. So We joined. My son was already far ahead of me, so I got one of the sterling members of the club to drop in on John once a week and see if he was making progress (in doing whatever it was that he was doing). He was.

It became apparent to me that if I had ever done anything right, getting my hands on that computer was the one thing. Most people would probably say that John was an intellectual kid. He uses his mind a lot. Not always the way I would, I guess, but where is a father who doesn't think his son or daughter shouldn't have better grades.

My wife and I don't like the idea that the kids let their brains rest over the summer. We let them know we expect them to have something going for themselves over the vacation period. John chose to go to computer camp at Western Illinois University in Macomb. He wanted to learn PASCAL. He went and he did. He came back from camp and went to the local library and brought home a couple of books of another language called LISP. He says LISP is being used for "artificial intelligence."

I have to ask myself: Have I created a monster?

Summer camp was fine, and I would recommend the program at W.I.U. to any parent who wanted to help a kid that was interested. I'm not so sure I want to drop the loot that it is going to take to come up with a PASCAL language system for the Apple.

Perhaps the way to express the problem (is it a problem?) is that he has begun studying the computer rather than using it. Getting the computer solved the riddle of how he was going to hold a pencil...BUT...we have just lost control. Did fathers feel this way before computers?

I suppose we will spend more money that I would like on various bolt on and/or plug-in accessories for the apple in the next three years. The thought that he may take a three or four thousand dollar piece of equipment out of the house to college gives me a chill. I don't know if I can put this equipment in the hands of someone interested in "beer busts, and blanket parties." A professor at my old alma mater (a college of less than a thousand students) says that there is a group of students already on campus who have brought their microcomputers with them to school. Furthermore, the college is encouraging them by trying to figure out a method to allow the students to reach the college computer system by modem.

Well, I haven't created a monster. There can't be any doubt though that I have started something that is already well beyond my control. Reviewing this process leaves a certain aftertaste of inadequacy. I don't know if I can even contribute any more than I already have. Granting that initial freedom of growth and exploration that was made with the introduction of the Apple to our household was like the last step at the end of the diving board. Natural forces have now taken over. Whatever the future holds in store as far as the computer is concerned, is now pretty much in the hands of John and his own ingenuity.

Judging from the current news, and articles in the magazines, I'm not the only one who feels this way. The Professor at my alma mater was a little puzzled about just what to do about guiding students with their own computers. Apples have now supplemented the Pets they had last year at the local high school. John taught one of his teachers about a program line editor he uses at home. Teachers teaching

students, I understand. Students teaching teachers is something I will have to get used to.

When John and I go to the NIAUG meetings the thought occurs to me that with all of these adults around, there must be a lot of kids somewhere. A few of them show up at the meeting, but they are largely intimidated by the adults who are doing their own thing. Nothing wrong with that. I just wonder, how many adults are racing just to keep up with the kids as I am?

This is it Terry. Just my thoughts on the computer in our house. It is out of control, but doing well. If this is what you are looking for in the way of an article, please use it. Change it in any way you see fit to suit your purposes...or don't use it at all.

I admire the work you, and our club officers are doing. Since I go to work at 0230 I am out of step with the rest of the world as far as participating in evening meetings. I hope this will help repay NIAUG for everything it has done for us.

* * * * *

DIGESTING THE APPLE

(WITHOUT SWALLOWING THE SEEDS)

by Robert D. Steinberg

No this is not another column dealing with the ins and outs of the internal (infernally?) machinations of the APPLE. What this column will hopefully develop into is a view of the APPLE and its associated world from the perspective of a new owner/user. I hope that the more experienced NIAUG members will bear with me as I attempt to share some of my thoughts, experiences, frustrations, and delights with those other members who, like me, are just beginning to explore the wide, and often confusing, world of the APPLE. Actually, I hope that the "senior" members will keep a close watch on this column in order to correct any errors I may make, and to clear up any areas of confusion that may plague me. I am sure that at least some of the things giving me trouble are also confusing to a few of the other "novices".

Prior to the purchase of my APPLE in mid-July, I had had no hands on experience with microcomputers although I was an experienced FORTRAN programmer. Now, six weeks later, I have a 48K APPLE II+ with one DISK II, a 12" green screen monitor, and as of yesterday, a NEC 8023A dot matrix printer (love that Greek character set!) with Grappler+ interface. I am writing this on a Southwestern Data Systems' text editor, THE CORRESPONDENT, that was being closed out at the local store where I bought the computer. I have lofty plans regarding future hardware and software purchases which, due to the finite size of my resources, will have to be put on the back burner for the time being. But more about that later. First, how did I, an otherwise sane individual, come to the conclusion that the APPLE was right for me?

Although I had no business applications in mind, there were enough other things I wanted to do to justify (in my mind) the purchase of a personal computer. I wanted to be able to develop some structural engineering programs and have some measure of independence from the University's computer for my research. I wanted to draft my Ph.D. thesis using a word processing program, and my wife and I have a large collection of technical and professional journals and magazines that we wanted to be able to index and abstract. There was also the usual amount of home financial information and other data that would be perfect for computer storage and analysis. The big question was which computer system would be best for our needs.

At first I was totally lost in the world of operating systems, 6502's, CP/M, and all the hardware and operational characteristics that a 'mainframe' user can blissfully ignore for the most part. (At least if all he's concerned with is getting his %\$ *& FORTRAN program running.) It took about a year of bugging the people at local computer stores and reading at least three computer magazines a week for me to get an idea of how to evaluate a computer system with regard to the applications that I foresaw. (Alas, I did not learn of NIAUG until shortly after I purchased my system.)

Well, after all that research and soul searching I came to the conclusion that a CP/M based system would be best for me and the graphics capabilities of the NORTHSTAR ADVANTAGE combined with its disk capacity

convinced me that it was my dream machine. After some wallet searching, however, I came to the conclusion that the combination of initial purchase cost and high price of most worthwhile CP/M software put the ADVANTAGE a bit out of my league. It was time to take another look at the APPLE with its skimpy disk capacity and forty column, upper case only, display.

After convincing myself that if I ever had the money I could upgrade the APPLE with 80-column boards, Z80 boards, hard disk, etc., etc., I came to the conclusion that the thing was so flexible that I couldn't lose. Besides, with all those APPLES already out there, and a ton of software to choose from, how could I go wrong? Little did I suspect that perhaps the biggest problem I would face after I got the computer home, would be to wade through the seemingly endless list of software and peripherals.

"Is there a big difference between the various 16K memory boards on the market? How about buying a non-APPLE second disk drive? What potential problems should I be aware of? How about a printer? Everyone seems to have an Epson, but the NEC and others look a lot better to me. Am I overlooking something important because of my novice status? Where do I find the answers to these questions? How can I avoid being burned by making an inappropriate purchase?"

Just about when I was reaching my breaking point in trying to answer these and other questions, a friend of a friend told me about NIAUG. I wasn't sure exactly what a user's group did, but it seemed to be worth attending a meeting to find out.

I joined NIAUG before the meeting started and went home with twelve disks from the NIAUG library. (I never knew that there was a way to run some Integer BASIC programs without a 16k card! See disk vol. 14.) After that one meeting, I knew that I would not regret my decision to buy the APPLE. The support and amount of information available to me through the user's group was fantastic. Best of all was the fact that here were all these other APPLE owners to learn from and eventually share my own experiences with.

I found out that for the money I want to spend, it doesn't really matter which 16K card I buy. I now plan to get a Kensington System Saver and hope to have Screenwriter II by the

time this reaches print. (Or should I look again at APPLE WRITER II?) I still haven't figured out what to do about a second disk drive. I'm reluctant to pay the high price for an APPLE drive but don't want to wind up regretting the purchase of a drive by another manufacturer because of service problems, or other problems. There's no rush to decide though. It will be a while before I can afford that second drive.

User's groups like NIAUG are perhaps the best part of owning an APPLE. (If I had purchased the NORTHSTAR I'd probably be struggling along with no one to turn to for help except the salesman who sold it to me.) Those of us just starting out should not be afraid to participate and get involved. After all, everyone was a beginner at some time, and eventually, we'll become old pro's. Unless we get actively involved in the organization, however, there may not be a NIAUG much longer.

This article represents my first effort to contribute something to the group which has already given me so much. I promise (well, at least I'll try) to keep my future articles from being so long-winded. Eventually, I hope to address topics such as "truth in advertising", which magazines and features I've found to be useful and which ones seem worthless, my initial attempts to decipher assembly language, priorities for software and hardware purchases, and the somewhat sad state that I perceive the microcomputer industry to be in vis-a-vis consumer support. Remember, this will all be from the point of view of a novice with the intent of sharing my thoughts with others. I don't claim to be an expert on any of these matters, but I feel that it would be worthwhile for everyone to be aware of how some of these areas appear to the beginner. I welcome your comments and suggestions regarding the topics I discuss, so that these articles can become a useful forum for everyone.

For now, though, is there anyone out there who can tell me if there is a trick to writing a program using random-access text files extending across multiple disks?

* * * * *

TALKS OF INTEREST

Dr Peter Lykos, will present a talk entitled "Advanced Placement Courses Based Computer Science and Computer Applications" 7:30pm, Reid Auditorium, Lake Forest Auditorium. This talk is of special interest to educators. Dr. Lykos has been involved in programs in computing for secondary teachers and students in Chicago since the 1960's and has held many offices in Chicago chapter of The Association of Computing Machinery. An outline of the Advanced Placement course in Computer Science and an overview of the reaction of the secondary and college communities will be discussed. He will also discuss a second Advanced Placement course in information science will be discussed and a well established BS in Chemistry approval program, conducted by the American Chemical Society since 1940 will be presented as a model for undergraduate BS programs in Computer Science and Computer Information Systems.

Captain Grace M. Hopper. USNR, will present a talk entitled "Possible Futures: Hardware, Software and People". Thursday December 2, 7:30pm Reid Chapel, Lake Forest Campus Dr. Hopper has a long and distinguished career starting with programming the first large scale computer ever built. She has received numerous awards and has authored more than 50 papers and articles on programming. Besides her many credentials and awards she is a very interesting speaker.

All talks are free and open to the public.

NEWS OF INTEREST

PARJON INFORMATION SERVICES

The more familiar users become with microcomputers the more they realize that the availability of backup and software is overriding importance. Much expensive equipment is gathering dust for the lack of good application software. Apple users are fortunate in having one of the largest application and software bases. There is still the problem of finding the information.

PARJON has introduced an abstracting service for the Apple computer. Their Apple has more publications dedicated to it than any other small microcomputer. In addition a number of papers dealing with a wide range of applications including, business, science and technology, education, and medicine are published in the specialist journals. APABS, as the service is called, gathers this vital information in the form of: full reference, a short summary, and a star marking. APABS is available as a disc version, a printed version, and combined disc and printed version. Further details are available from:

PARJON
14 Broadway
London SW1HOBH, England

- 1-.-.-----
A PORTABLE RELATIONAL DATA BASE MANAGEMENT SYSTEM FOR MICROCOMPUTER
:::G.FALQUET, D.PETITPIERRE, N.MAGNENAT-THALMANN, D.THALMANN

MICROPROCESSING AND MICROPROGRAMMING 1982 JAN VOL.9 NO.1 17-25. (HOLLAND E
****)

The design and implementation of a DBMS named Titan is described. The
system has been used on a 54k Apple II plus with three disc drives. The
program is in UCSD Pascal
- 2-.-.-----
CHARACTER GENERATION EXPERIMENT FOR A COMPUTER LOGIC COURSE :::D.Q.M.FAY
D.Q.M.FAY

MICROPROCESSING AND MICROPROGRAMMING 1982 FEB VOL.9 NO.2 23-28. (HOLLAND E
****)

Undergraduate experiments with character generation ROM are dealt with.
Circuit details are given for a dot matrix display system, and experiments
to examine the way in which a video monitor is driven. An ITT 2020 with
UCSD Pascal was used.
Education
- 3-.-.-----
WHITHER THE HOME ROBOT? A PERSONAL OPINION :::A.GELLES

MICROPROCESSING AND MICROPROGRAMMING 1982 MAY VOL.9 NO.5 285-288 (HOLLAND
E **)

The future of home robots is discussed with mention of Apple II as an
example of a personal computer capable of supporting robotic research.
- 4-.-.-----
DATABASE DESIGN- A TUTORIAL IN REPRESENTATIVE APPLE SOFTWARE :::J.C.BLAIR JR

DATABASE 1982 JUNE VOL.5 NO.2 72-79 (USA E ***)

A short outline is given of, DB Master, Supertext II, Visiterm and
Visiplot. photographs of representative 'screens' from these four programs
are given.
- 5-.-.-----
SMALL COMPUTERS IN THE ARMY- AN APPLE A DAY TO KEEP THE SOVIETS AWAY
:::W.P.SCHNEIDER

SIGNAL 1982 FEB VOL.36 NO.6 39-42 (USA E ***)

An outline is given of the use of the Apple II plus and supporting
hardware in the Theater Target Analysis and Planning (TAP) system

- 6-.-.-----
TRACKING THE CREATIVE CYCLE WITH A MICROCOMPUTER :::M.L.G.SHAW, B.R.GAINES

INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 1982 JULY VOL.17 NO.1 75-85
(UK E ***)

Kelly's personal construct psychology is used as the basis for an analysis
of the educational process and system, with emphasis on the determination
of the knowledge state. Programs for the interactive elicitation and
cluster analysis of personal construct system on a Decsystem 10 mainframe
have been extended and transferred to the Apple II as an integrated suite
of programs for routine use (Planet)-CAI,CAL
- 7-.-.-----
A MICROCOMPUTER-BASED SYSTEM FOR INSTRUMENTED IMPACT TESTING :::R.J.CRAWFORD
N.V.MCFERRAN A.AL

JOURNAL OF TESTING AND EVALUATION 1982 MAY VOL.10 NO.3 121-126 (USA E ***)

A modification to an Izod type impact testing machine is described, which
permits fully instrumented tension impact tests. A load cell in the base
of the machine sends the impact signal through a charge amplifier to a
transient recorder. The stored data is processed by an Apple with one disk
drive and two parallel bidirectional interfaces using Versatile Interface
Adaptor (VIA) IC's type MCS6522
- 8-.-.-----
IMPROVING SIMULATION PERFORMANCE ON THE APPLE II MICROCOMPUTER :::J.R.RANDALL

SIMULATION 1982 JAN VOL.38 NO.1 26-27 (USA E ***)

A short note on implementing biomedical teaching simulations. Examines the
speed of running a simulation of the cardiac action potential model of
Beeler and Reuter, and the effect on speed of compiler, interpreter,
software floating point and hardware floating point. -- Hayden Applesoft
Compiler, On-Line Applesoft Expediter II Compiler, CCS 7811 Arithmetic
Processor.
- 9-.-.-----
DISC TO TAPE BACKUP UTILITY :::R.MERTON

MICRO:THE 6502-6809 JOURNAL 1982 JUNE NO.49 9-11 (USA E ****)

A short machine language - integer basic listing is given, which will dump
an entire 16 sector disk to tape in about 15 minutes. Four disks can be
stored on a C60
- 10-.-.-----
A LOW-COST DIGITIZER FOR THE APPLE :::J.SINNETT

MICRO:THE 6502-6809 JOURNAL 1982 JUNE NO.49 40-43 (USA E ****)

A method is described which uses a sheet of half-reflecting plastic
carefully aligned in front of the monitor. A listing of the required
program is given.

UNCLASSIFIED ADSFOR SALEHome/Small Business System:

APPLE II 64K, Disk w/c, game paddles, RF mod., C. ITOH 8510, monitor (B/W), software. The software consists of PFS(new version), VisiCalc, Screenwriter II. They are all the latest editions. There is also 2 Compu-Math programs and some games that come with it as a package. The system is 8 months old and I am going into a larger business system associated with the business. It is in excellent condition. I would like to sell it as a package, but I can separate the printer...asking \$2850 or offer.

Jay Larkin
1758 Black Road
Joliet, Il. 60435
(815)725-1137

FOR SALE

APPLE II+ w/48K, Disk II w/DOS 3.3, Leedex Video 100 Monitor. All system documentation, some programs. \$1,350.00. Call Grant Shaw, 893-5408 after 5 PM.

FOR SALE OR TRADE

Dna/Avnet Serial TTY Printer \$100, Conrac Monitor (no case) \$40, several cases of paper tape (fanfold and rolled) make offer, Software (original disks and documentation): The WORD Processor (by Bible Research Systems) \$100, AgDisk:Financial Management Series One \$75, call Larry 858-7287.

HELP WANTED

MICRO-PROGRAMMERS, \$30-40,000 salary, profit-sharing, substantial bonuses. We require two years of Assembler language programming experience on micro-computer systems (Intel, Motorola, Zilog, 6502) with knowledge of REAL TIME applications and preferably graphics.

We offer an excellent career path, interesting work on state-of-the-art hardware and graphics systems and a good working environment. Write or call John Doby, 6055 North Lincoln Avenue, Chicago, Il, 60659. 312-973-6969.

INVESTMENT SIG

Meets on the third Thursday of every month at 7:30P.M. at:

Morton Grove Community Church
8922 North Austin
Morton Grove, Illinois

(On Austin Avenue one block north of Golf road). For additional information call Jim Bradshaw at 881-7000.

BUSINESS SIG

Meets on the third Saturday of every month at 10:00 A.M. at:

Mount Prospect Public Library
10 South Emerson Street
Mount prospect, illinois

(One block East of the intersection of Elmhurst Road and Central Road)

November 20th Meeting, Introduction to VisiCalc.

For additional information call Jack Gratz at 448-6548.

Pascal SIG ———Ken Nestle 312- 620-7745

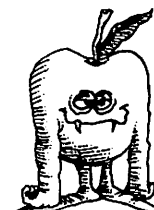
Modem SIG ———Terry Cronin 312- 289-6392

ASSEMBLY LANGUAGE SIG MEETING

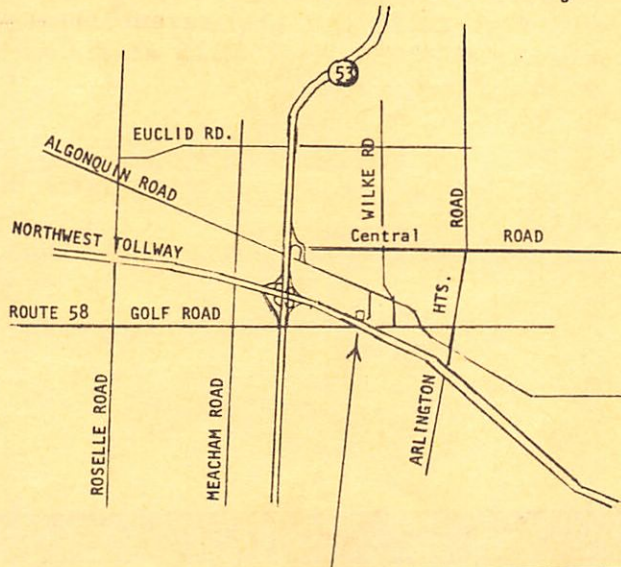
The Assembly Language SIG will meet on Sunday December 5, 2 pm at the Palatine Library. For more information call Ron Metzger 885-1979 or Don Larson 547-1467.

NIAUG NEANDERTHAL APPLE PINS

These pins are available for \$3.00 each at the treasurer's table.



CLUB NEWS



NIAUG PLANNING MEETING

1st Tuesday Following Monthly General Meeting, 7 PM

APPLE COMPUTER, INC.
DISTRIBUTION CENTER
5655 MEADOWBROOK
ROLLING MEADOWS, ILL.

577-3600

Access to Meadowbrook is only available
from Algonquin Road.

NOVEMBER AGENDA

Nov. 13, 1982.

Octobers meeting will be held in
Building A.

- 10:00-10:15 am Opening Remarks
Eric Stral
- 10:15-11:00 am All you want to know
about computer software
but were afraid to ask
Bill Osmer
- 11:00-11:30 am Statement of the month
Guy Lyle
- 11:30-11:45 am Break
- 11:45-12:15 pm Business Graphics
Eric Stral
- 12:30-12:55 pm Mr Apple
- 12:55- 1:00 pm Closing Remarks

HAMS (non thespian variety)

HAMS with Apples will be meeting again at
9:00am at McDonalds on Algonquin near Harper
prior to the NIAUG monthly meeting. This is a
small informal group that has no formal
leadership but that may develop into a SIG
eventually.

Also the Apple NET meets on 14.329 on Sundays
at 8:00pm. The last Sunday of October will be
devoted to CW and RTTY programs.

1982 NOVEMBER 1982

SUN	MON	TUE	WED	THU	FRI	SAT
FULL MOON 1	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	LAST QUARTER 8	NEW MOON 15	FIRST QUARTER 23	

PLANNING MEETING

INVESTMENT SIG

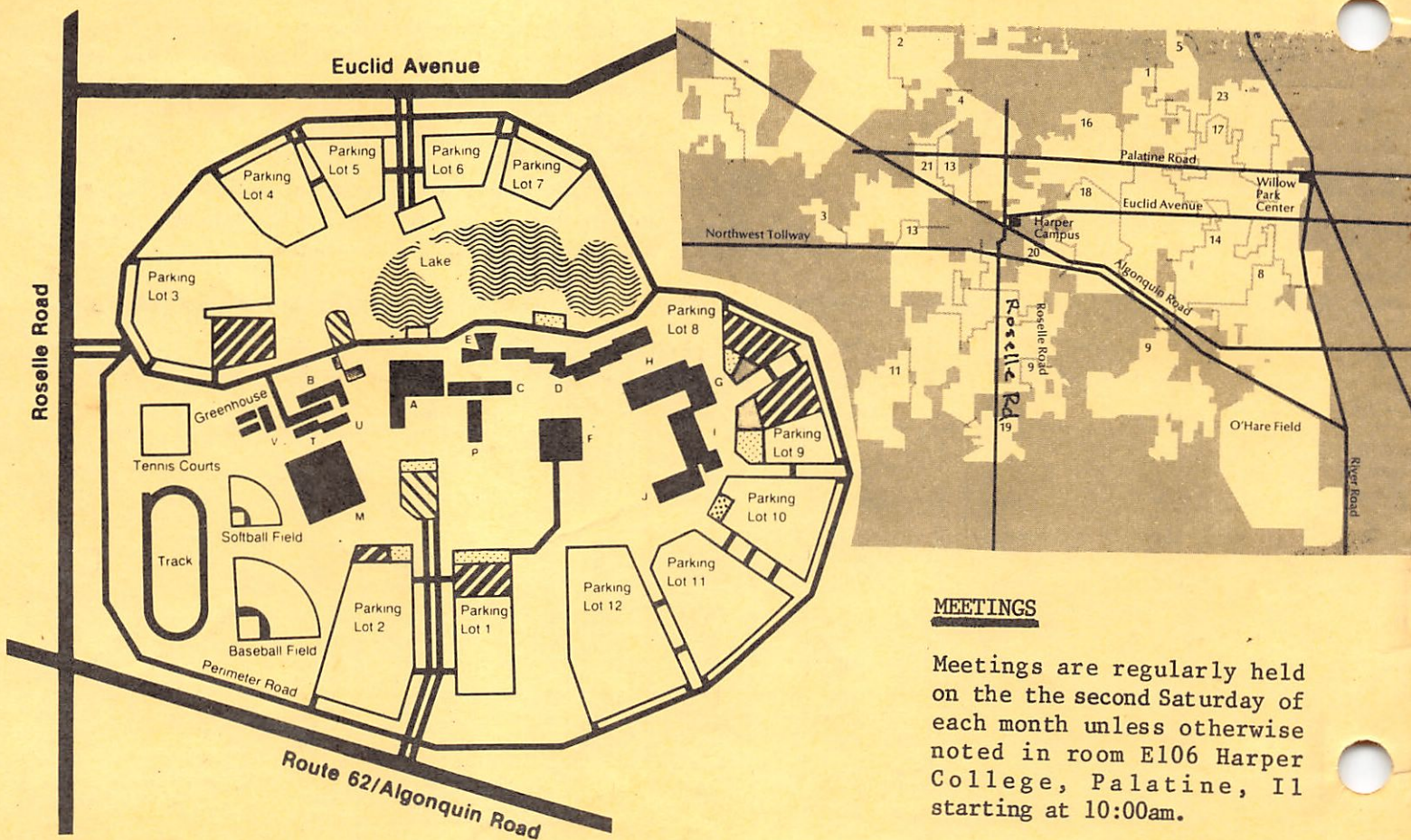
GENERAL MEETING
BUS SIG

THE HARVEST NEWSLETTER
N.I.A.U.G.
1015 S. RIDGE AVE,
ARLINGTON HEIGHTS, IL.
60005.

BULK MAIL
US POSTAGE PAID
PERMIT 170
MT PROSPECT IL

MICHAEL L. ROBINS
R.R. #1 BOX 53B
NEW LENOX, IL 60451

POSTMASTER - CONTENTS DATED
- RETURN POSTAGE GUARANTEED
- FORWARDING REQUESTED
- ADDRESS CORRECTION REQUESTED



MEETINGS

Meetings are regularly held on the the second Saturday of each month unless otherwise noted in room E106 Harper College, Palatine, IL starting at 10:00am.